



IP-FP6-015964

AEOLUS

Algorithmic Principles for Building Efficient Overlay Computers

Deliverable D6.2.10

New components of the Overlay Computing Platform

Responsible Partner: University of Patras (EL)
Report Preparation Date: February 2010

Contract Start Date: 01/09/05 Duration: 54 months
Project Co-ordinator: University of Patras (EL)

Deliverable D6.2.10 consists of all software components produced during the last reporting period of AEOLUS. These are either functionalities described in Deliverable D6.1.2 (Overlay Computing Platform: Design Report) or additional functionalities identified during the third year of the project and promised in the updated implementation plan. The latter include:

- publish/subscribe functionalities, the JXTACh rendezvous service, distributed peer-to-peer storage, and mechanisms for social choice functions in relation to SP2,
- load balancers and scheduling functionalities, the microbenchmarking suite, and replication services in relation to SP3,
- secure searching of outsourced data, non-malleable CAPTCHA for combatting DDOS, and the prototypes Trust-X and SybilGuard in relation to SP4, and
- high-level (i.e., mobility prediction and control, clustering/grouping, time synchronization, and code update) and low-level functionalities (i.e., buffering services, high-level network programming, and SQL-like functionality) in relation to SP5.

In addition, we have implemented some functionalities/prototypes that are not included in the above list. In the following, we present a very brief summary of the available implementations.

Functionalities related to SP2: Publish/subscribe functionalities for the overlay computing platform enhanced with ranking capabilities have been implemented by UOI. UOI has used a ranking mechanism based on user preferences, so that only top-ranked events are delivered to each user. Since many times top-ranked events are similar to each other, the diversity of delivered events has also increased. Concerning the ranking of events, UOI has implemented a number of different delivering policies for forwarding ranked events to users, namely a periodic, a sliding-window and a history-based one. UDRTV has completed the implementation of the JXTACh rendezvous service. JXTACh replaces the “lightweight” distributed hash table implementation of JXTA with a real Chord DHT. In particular, JXTACh replaces the JXTA rendezvous peer view with the Chord finger table applied to the JXTA rendezvous network and the JXTA limited range walker mechanism with a *Chord walker*, that has the same task of the JXTA walker, but follows the Chord algorithm instead of linearly forwarding queries. This results to improved performance of discovery, publication, and query for advertisements. UOP has implemented social choice (voting) functions as services of the AEOLUS testbed. This service allows for the definition and the execution of an election among users. Several voting schemes have been implemented (Plurality, approval, Borda count, and Tideman) in which the users (voters) either rank all alternatives (candidates) or approve a set among them. The outcome is then computed by applying the selected voting scheme on the users’ preferences. These

functionalities have been integrated into the AEOLUS testbed. In particular, JXTACh is used as an alternative for JXTA.

CNRS has studied long-term reliable storage systems over peer-to-peer networks, where the goal is to achieve high reliability and fault tolerance at low cost. The key concept of such systems is to encode the user data into a set of redundant fragments and to distribute them among peers. CNRS has also performed experiments to analyze P2P storage systems using the local nodes of the AEOLUS testbed based at CNRS. Moreover, CNRS has implemented a scalable peer-to-peer based Carpool information system that exploits resource discovery functionalities offered by the Arigatoni protocol and his evolution, the Synapse protocol. CNRS has also proposed Open-synapse, an open source implementation of the BabelChord, written in Java, which is a variant of the Synapse protocol for information retrieval over the inter-connections of heterogeneous overlay networks.

Functionalities related to SP3: Additional features of the AEOLUS testbed core have been implemented by UPB. These include distributed load balancing algorithms that are able to handle the different processor speeds in a heterogeneous network by using distributed heterogeneous hash tables (DHHTs). Furthermore, novel distributed algorithms for clustering the PUB-Web network according to bandwidth and latency have been implemented. UNIPD has completed the implementation of the microbenchmarking suite. It is now integrated with PUB-Web so that scheduling results in more balanced allocations, while matching process preferences against node characteristics. UOP has upgraded the scheduling functionalities for OCP applications by providing a complete API that is now available for application and is used by the AEOFORGE application. The new scheduling methods implemented include truthful scheduling mechanisms for unrelated machines. MPII has integrated in the AEOLUS testbed a new version of the P2R2 method for near-optimal dynamic replication. P2R2 considers unstructured overlay networks and aims to maximize the probability of successful search by means of bounded random walks.

Functionalities related to SP4: UNISA has continued working on cryptographic primitives for protecting data privacy. They have implemented secure searching on outsourced data (SSOD) that allows storing data in an encrypted form on remote servers so that it is possible to perform search operations without the requirement to download (and decrypt) the entire data set. UNISA has also implemented non-malleable CAPTCHA for combatting distributed Denial of Service attacks (i.e., attacks against a network service where an adversary abuses the service in such a way that it becomes unusable for other users) that is resistant to man-in-the-middle attacks. These functionalities have been integrated in the AEOLUS testbed. In particular, SSOD is used by the AEOFORGE application. K.U.Leuven has improved upon SybilGuard's performance with respect to scalability and communication overhead. Furthermore, the service is now easier to adapt to a real world social network. UOI/Insubria have completed the development of a TrustX

prototype (Deliverable D6.2.11) supporting the XRNL language in Java and JXTA. The implementation consists of a JXTA membership service and of the infrastructure needed to actually verify the disclosure policy required by the peer and the group. The initial TrustX prototype has been extended with further functionalities, like negotiation crash-recovery, support for long-running negotiations, and resource negotiations between peers' groups. In order to support such features and to ease the integration of future possible extensions, the code of the Trust-X prototype has been thoroughly refactored and in large parts completely rewritten.

Functionalities related to SP5: During the final period, all promised functionalities (both low-level and high-level) have been integrated to the testbed. In terms of low level services, CTI has implemented and integrated the following functionalities: i) mobility prediction and control functionality that uses mobility prediction to improve the performance of the communication services in the case of mobile nano peers and also allows the gateway peers to move in order to collect the sensed data instead of passively waiting for the data to be propagated to them by the network, ii) clustering/grouping functionality that provides methods to form dynamic, ad hoc group of nodes according to the requirements of a task and resource availability. Furthermore, it maintains information concerning the nodes of a group in the presence of mobility in a consistent and persistent manner, iii) time synchronization functionality that maintains local clocks in each node of the network and keeps them all synchronized (absolutely or relatively) to a master clock so that nodes can provide accurate reports, topology control and energy saving schemes, iv) and code update functionality that is responsible for the dynamic remote reprogramming of all network nodes. This functionality also provides the framework for building mobile agent programs that propagate through the network.

High-level functionalities that were integrated include: i) buffering services that enable the overlay network to handle intermittent network disconnections that are possible due to several reasons (e.g., poor quality links, environmental noise, etc.). This buffering is applied to all data, i.e., both data and control messages. ii) SQL-like functionality that is responsible for the organization and storing of data after sensing actions, how program interfaces to the sensor database look like and how queries are processed and served in an efficient manner. iii) high-level network programming.